Fig. 1
An efficient merger
procedure
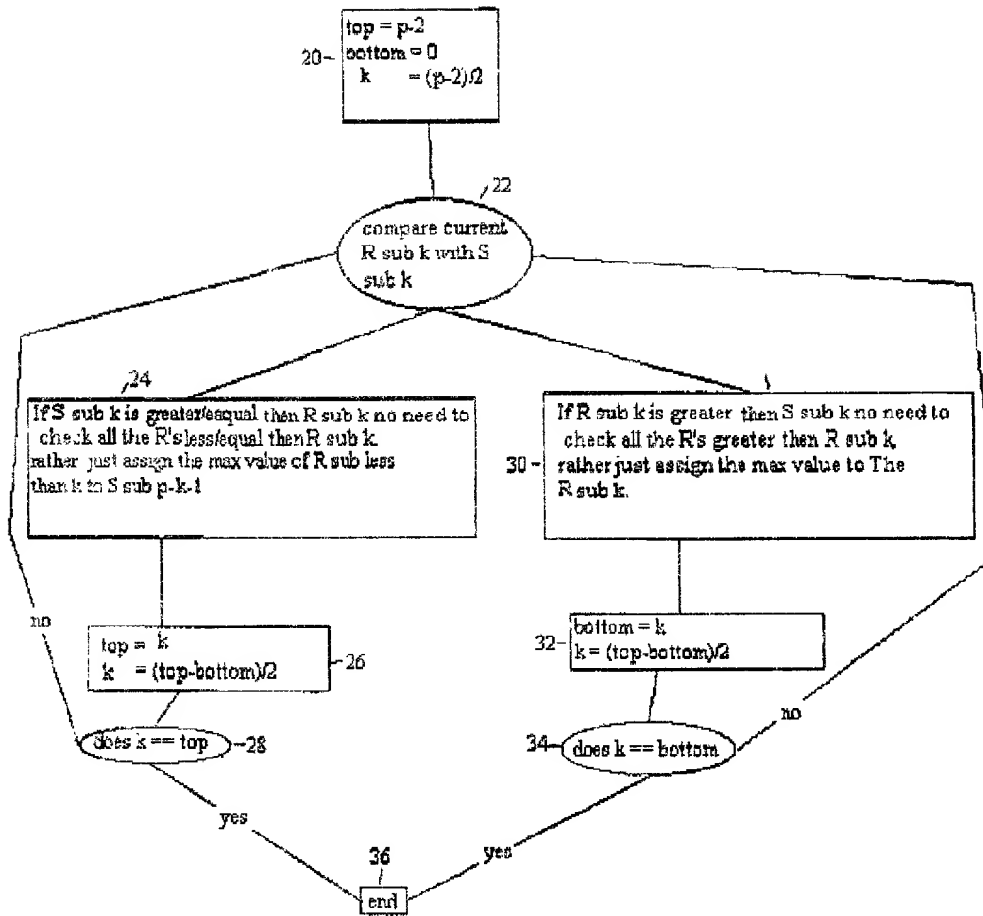
Fig. 2

an efficient
pre-
processing
procedure

do for all segments defined by the GWVH algorithm

define $q = (p/2) + [(p \bmod 2)/2]$ — 38

compute all s sub k for k=0..q-1 — 40

compute all r sub k for k = q...p — 42

compare s sub q-1 with r sub q — 44

r sub q is larger   s sub q-1 is larger   48

46

output:
r sub q-1 = r sub q-2 =

= r sub 1 = r sub q,
continues to compute
s sub q...p for the rest of this
segment

output
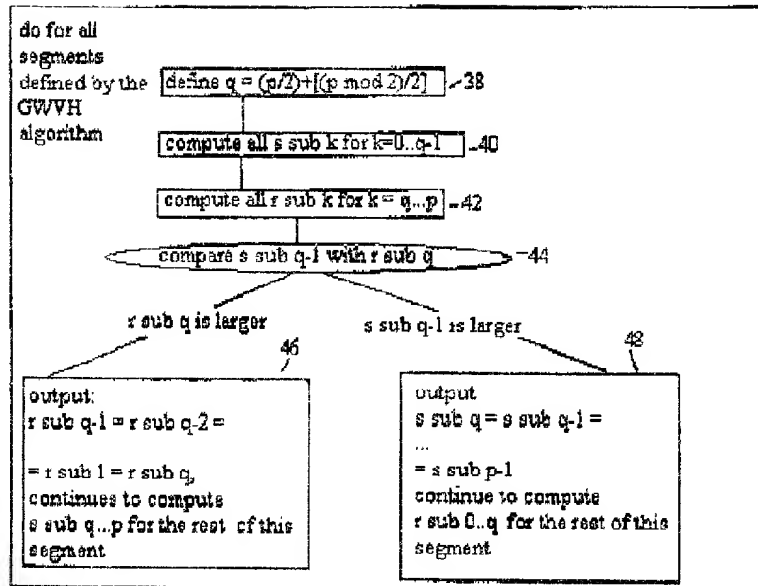s sub q = s sub q-1 =
...
= s sub p-1
continue to compute
r sub 0..q for the rest of this
segment

Fig. 3

an efficient
opening filter

```
compute the max filter, keeping track of          ~50
the segments, using the improved
GWVH algorithm
```

62

```
pre-processing the data with the min
filter, the min filter having access to the
segmentation from the max filter
do for all segments

    assign the first value of the end of each       ~52
    decreasing segment to the entire segment

    binary search on an increasing segment to find
    the first place where the value of the current   ~54
    datum X sub i is greater than or eaqual to the
    value of the datum at position X sub i+p

    for all data befoe X sub i, in this segment,     ~56
    the min is the value of that segment

    for all datum after X sub i, in this segment the  ~58
    min is the value of X sub i+p
```

```
end of pre-processing, continue with the efficient merge of
fig 1                                              ~60
```